DOCUMENT RESUME

ED 050 609	24	EM 008 986

AUTHOR	Shapiro, Gilbert
TITLE	Programming for the Facile Use of the I.B.M. 360 Computer As a Laboratory Instrument in the Social Sciences. Final Report.
INSTITUTION	Pittsburgh Univ., Pa.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-8-A-009
PUB DATE	Aug 70
GRANT	OEG-1-8-08AC09-0037-057
NOTE	12p.
EDRS PRICE	EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS	College Mathematics, Computer Assisted Instruction, *Computer Programs, *Electronic Data Processing, Programing Languages, *Social Sciences, Statistical Analysis, *Statistics

ABSTRACT

A system was envisioned which would enable the graduate-level social science student to use a computer to construct statistical procedures according to theoretically significant formulas, rather than to provide output in the most efficient manner. It was hoped that this system would allow the student to spend more time learning to manipulate statistical data and less time learning to program the computer. Thus far, a number of sub-routines have been programed which provide simulated data and which allow the user to modify this data and to construct standard research statistical techniques. Some work has been done on a control program which will consist of a series of "pre-compiler" statements to be incorporated automatically into student programs. This control program will provide the technical "overhead" programing and relieve the student of the technical concerns. This report stipulates the design criteria and the programing strategy for the system. It describes briefly the procedures already compiled and the present state of the control program. A proposal for future work to complete the system is presented. (JY)

FINAL REPORT
Project No. 8-A-009
Grant No. OEG-1-8-08A009-0037-057

PROGRAMMING FOR THE FACILE USE OF THE I.B.M. 360 COMPUTER AS A LABORATORY INSTRUMENT IN THE SOCIAL SCIENCES

Gilbert Shapiro
University of Pittsburgh
Pittsburgh, Pa. 15213

August 1970

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

FINAL REPORT

Project No. 8-A-009
Grant No. OEG-1-8-08A009-0037-057

PROGRAMMING FOR THE FACILE USE OF THE I.B.M. 360 COMPUTER AS A LABORATORY INSTRUMENT IN THE SOCIAL SCIENCES

Gilbert Shapiro

University of Pittsburgh

Pittsburgh, Pa. 15213

August 1970

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

TABLE OF CONTENTS

SUMMARY

The objective of this project has been to provide a system of programs which would enable teachers of social statistics to offer their students the opportunity to study the behavior of statistics under varying conditions of input. Previous uses of computers in teaching statistics have been usually based upon packaged programs which provide only output parameters for standard techniques. The present system, organized in modules, is designed to enable the student to construct statistical procedures, according to theoretically significant formulas, rather than to provide output in the most efficient (and theoretically meaningless) manner.

The strategy adopted has been to attempt to provide the student-programmer with a simplified version of the PL/1 language, specifically adapted to statistical experiments. Programming has been completed on a number of sub-routines which provide simulated data, and the capacity to modify these data at will, as well as modules from which can be constructed standard research statistical techniques. In addition, to simplify the writing of main programs calling upon these sub-routines, work has progressed on a series of "pre-compiler" statements to be incorporated automatically in student programs, which provide the technical "overhead" programming, and relieve the user of most technical concerns. These packages of statements, which serve the function of a "control program" have not been completed and debugged. Future work will include the completion of this program, the completion of a manual for the use of the system, and the "debugging" of the entire educational package in classroom experience.

INTRODUCTION AND RATIONALE

This project grew out of the experiences, and the difficulties, of the principal investigator in teaching statistics to social science students at the graduate level, attempting to utilize the computer as a laboratory instrument. The essential educational objective was to supplement basic mathematical presentations of statistical techniques with demonstrations of how statistics behave under varying conditions. While conventional approaches to the presentation of statistical methods through mathematical derivations remain of the greatest importance, there are many reasons why they are insufficient under contemporary circumstances. First, many students, particularly in sociology, arrive in the class-room with little mathematical background, and considerable fear of symbolism. Second, even when derivations are grasped, teachers have long recognized intuitive elements lacking in a course with a strictly mathematical approach; we generally feel that we want students to have a "feel" for the techniques as well, and for this reason laboratory exercises with paper and pencil, or desk calculators, have been a normal part of the pedagogy of statistical training in the social sciences. Finally, the computer (in principle) opens up great new possibilities for student experimentation with statistics, eliminating the drudgery of hours of feeding input figures to a calculator to get merely a single, isolated result. In fact, a multiplicity of such calculations is often required to demonstrate some statistical principle with a range of input distributions, or where some principle is "generally" true, over a mass of randomly generated distributions.

We said that the computer opens up such possibilities "in principle," because, particularly in the early days of computer technology, grave practical difficulties arise in the attempt to use computers for such purposes. Computer languages have been as forbidding as the mathematical symbolism of statistical derivations. Vast amounts of class-room time, and of the student's studying time, are often devoted to learning programming languages, and the rules for the use of operating systems. Courses which were supposed to teach statistics ended up largely devoted to computer techniques, and the students were deprived of much of the statistics curriculum. Programming errors often became the major pre-occupation of students and teachers, rather than statistical principles; the means became the end.

A number of systems were developed to deal with these difficulties by providing a simplified language which could be used for statistical computations by those without a knowledge of programming. These, in general, were designed primarily for the use of researchers, rather than students. Researchers require

the <u>results</u> of as wide as possible a variety of statistical computations, with the greatest possible machine efficiency, and with some attention to simplicity in the rules by which the user instructs the program. Systems, such as DATATEXT and M.S.A. developed at Harvard, have served definite functions in the education of social science students in statistics, but they compromise the educational function with research purposes in many ways which distinguish them from the system developed by this project, described below. Fundamentally, they assume the existence of a set of data, and of an established research method, generally some multivariate technique, and provide the user with the parameters which constitute the essential <u>outcomes</u> of the application of an established method to an existing data set. For the purposes of a researcher, these services suffice, and students can learn much from their use.

However, since our objective was exclusively to demonstrate to students some principles of statistical method, the system was designed to meet the following criteria:

(1) It must provide the capacity to create and modify data sets at will (so that the behavior of statistics can be studied under varying conditions of input).

(2) It should provide any inter-mediary results that might be of interest for study, and not merely the output parameters. (This criterion generates a requirement that the system by <u>modular</u>, meaning that each step in computation that has conceptual interest can be called upon separately, and the <u>development</u> of the final results can be studied, as well as the ultimate product.)

(3) The pedagogical requirements of demonstrating statistical principles often conflict with the technical requirements of maximum computational efficiency which properly guide the development of systems designed primarily for research use. For example, while it makes <u>conceptual</u> sense to compute a Pearsonian correlation coefficient by finding the mean of the cross-products of the standard scores, this procedure is far slower than the use of available conventional computational formulae. On the other hand, it would be very easy for a student with no understanding at all of a correlation coefficient to call upon a program which computes it efficiently. In this system, we have provided modules which can be called successively to provide theoretically meaningful computations, so that the student can enter into the exercise of <u>constructing</u> the statistic. After he has trained himself in this way, more efficient modules are available to produce the same result with greater computational efficiency, but without the pedagogical values of step-by-step methods.

(4) A final consideration entering into the plan for this project was the importance of the publication of the results, in a form that could be immediately utilized by teachers at a large number of colleges and universities. Systems developed in the past which did have some of the modular characteristics required for our purposes, including those at Dartmouth, M.I.T., and the Systems Development Corporation, were all machine-specific. While these programs were studied for useful ideas, aside from other limitations, all are available only for use on the machines on which they were designed, at their home installations. The present system, developed in the language PL/1 is designed to be immediately used on any model of the I.B.M. 360 series which operates under O/S, the most widely diffused set of computer systems at the present time. (The system 370, announced for distribution by the Corporation at the time this report is being written, is said to be fully compatible with programs developed for the 360.) Besides planning for a system which could be installed in the most widely used machinery, the programming has utilized throughout the language capabilities designed for greatest flexibility, such as controlled storage, adjustable array boundaries and logical file names, so as to enable us to adapt to any installation's unique complex of equipment within the I.B.M. 360, and 370 range.

METHODS

In order to provide the user with the utmost flexibility and ease in instructing the system, our programming strategy has utilized the pre-compiler features of the PL/1 language extensively. This is a processing stage in which program statements are taken as input, and modified according to special instructions provided by the programmer. By this method, the student's instructions can call upon functions and subroutines already compiled and stored on disk, but can also call upon collections of program statements, (such as variable declarations) which are not in the form of "procedures," that is, subroutines or functions, but are merely slices of a main program, in source language. In addition, the student can use any variable names he wishes. In effect, the student is writing a main program, but because he can include collections of statements already written, and called up from the disk by a simple card, as well as call upon subroutines and functions, the programming knowledge required for the production of relatively complex programs is vastly reduced without the loss of flexibility usual in program packages. This advantage is purchased at the cost of a small amount of computer time --that required for the pre-compiler phase processing, and the compilation of a main program. Experiments indicate that even for relatively extensive and complex programs this will not require more than three minutes on the Model 360/50, and for typical programs less than a minute.

RESULTS: COMPILED PROCEDURES

We will begin by describing the PL/1 "procedures," i.e., the subroutines and functions, which have been written and debugged, and then proceed to discuss the pre-compiler processing features of the system.

1. Data Generation Procedures.
    - A. Nominal Data. The student may specify the number of categories in each of up to four variables, and, if he wishes, the values in the cells or the marginals. In the latter case, cells are filled by random assignment within the limits set by marginals.
    - B. Interval Data. Random variates, and any number of rectangularly or normally distributed random arrays, are produced with mean, standard deviation, and size of array specified by the student.
    - C. Distribution Modifications. An array once produced may be modified at will, by the addition of values chosen by the student, or by the elimination of all those within a given range. Since the sequence of operations is entirely in control of the student, he can make repeated computations on the same statistic after each of a succession of modifications of the distributions involved.
    - D. Correlated Variables. Two variables are produced whose zero-order correlation is equal to a value provided by the student. The variables are normally distributed.

2. Computational Procedures on Nominal Data.
    - A. Percentages. Computed in either direction (or both).
    - B. Chi-Square values with expected frequencies.
    - C. Phi-coefficient and Kruskal-Wallis measure of strength of relationship.

3. Computational Procedures on Univariate Interval Data.
    - A. Conversion to ranks.
    - B. Conversion to standard scores.
    - C. Descriptive Univariate statistics: Mean, median, standard deviation, variance.
    - D. Frequency distribution within intervals selected by the student.

4. Computational Procedures on Bivariate Interval Data.
    - A. Parameters of regression equation, a and b.
    - B. Pearson zero-order correlation coefficient.
    - C. Expected or predicted values of dependent variable.
    - D. Residual Scores.

5. Computational Procedures on Multivariate Interval Data.
   A. Partial correlation--first order and second order.
   B. Multiple correlation--two independent variables, and three independent variables.
   C. Extraction of first principle component from three variables.

6. Ordinal Data: Computation of gamma.

7. Analysis of variance: one-way and two-way.

THE CONTROL PROGRAM

The procedures described above are designed to enable the student to easily perform tedious operations by calling upon them by name, in any order he wishes, in accordance with rules communicated in a manual. Since he is, in fact, writing a PL/1 main procedure, all of the facilities of that powerful language are available to him, but we do not envision the typical student as having a knowledge of the language. Our control program is designed to simplify the PL/1 programming language, so as to eliminate all tedious overhead programming, with a minimum of reduction in the flexibility of possible computations. Two features of the PL/1 compiler enable us to do this: its default features, which often enable one to ignore various requirements of communication and specification, particularly of data types, and its pre-compilation phase. The latter enables us to provide the student with the facility to include in his main program statements already written and residing on the disk, as well as to change the name of variables at will. We believe that these features enable the student, though ignorant of programming, to have a larger share of the benefits of flexibility in instructing the computer enjoyed by programmers, than when using packages fed by execution time control cards.

The functions normally performed by a control program are largely performed by a combination of Job Control Language statements and an "Overhead" package of source language statements to be included in every student program by virtue of a card requesting these statements in the pre-compilation phase of his run. Unfortunately, because of difficulties with the PL/1 package as provided by I.B.M., and as implemented in the local installation, we have not been able to complete this work as of the present date, but we do expect to be able to do so shortly.

CONCLUSIONS: PLANS FOR FUTURE WORK

For many reasons, we have not been able to complete a fully operational system during the time period scheduled for this project. We have been working in a computer language which has been developing, with many difficulties due to temporary errors in the compiler and unsupported features, which have only gradually disappeared. In addition, because of unusual delays in funding, the project has been moved, along with the principal investigator, with the attendant difficulties of finding new staff and adjusting to new computer installation conventions and facilities. Under the present grant we have, nevertheless, pushed the effort very close to the point at which useful programs can be provided to the profession.

1. We intend first to complete programming of the control system, so as to make the system operational.

2. Work has begun on a manual, whose final version must await completion of step 1, to test its clarity of presentation with actual classes of students, and to find errors of omission that might be confusing to students.

3. Publication of the manual along with announcements of the availability of the programs to teachers through professional journals.

4. When a suitable time-sharing version of PL/1 becomes widely disseminated, the program will be adapted to this, far superior, mode of computer use in instruction.